Bachelor Thesis

Abdelrahman Mostafa

July 28, 2023

Abstract

This is my abstract.
This is my first LaTeX document.

Acknowledgement

Summary

Contents

List of Figures

Chapter 1

Introduction

speak about fluxana, and what it's doing. Then we after understanding what fluxana is doing, we start with the problem section.

1.1 Problem

What is the problem

1.2 objectives

how we want to tackle this problem.

1.3 Limitations of this work in the thesis

1.3.1 What I excluded from the thesis

1.4 take the person the through the whole paper structure briefly, (one papgraph)

1.5 Related works

what papers or researchs are made on this project

Chapter 2

Linux

Chapter 3

Robot Operating System — ROS

First reference [1].
Second reference [2].

3.1 MoveIt!

Chapter 4

Powder Dosage

explaining a bit about the powder dosage task.

4.1 Different parameters

4.2 Used Crucibles

Chapter 5

Task-Setup

5.1 Hardware

5.1.1 Niryo-Ned2

5.1.2 Gripper

5.1.3 frame

5.2 Software

Chapter 6

Experiments

6.1 What tests are made?

and what are the outcomes?

Chapter 7

Results

The last and final results.

Chapter 8

Conclusion

small description of what I have done.. What are my final findings and thoughts...

The future work, what to come.

Bibliography

[1] Anis Koubâa et al. *Robot Operating System (ROS).*, volume 1. Springer, 2017.

[2] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

Appendix A

Python Code with a fancy background

```
1
2  def Call_Aservice(service_name, type, request_name=None,
       req_args=None,should_return=None):
3
4      """Call a ROS service.
5
6      Parameters:
7      ......................
8      service_name: str
9      type: srv
10     request_name: None (srv)
11     req_args: None (dictionary) ex. {'positon': 210, 'id': 11,
       'value': False}
12     should_return ?: None (int) >> is set to 1, if you want to
       return the response of the service.
13
14     Returns:
15     ......................
16     If should_return is set to 1, the function is going to
       return the response of the service.
17     Otherwise, the function should only call the service to do
       a certain action with no return.
18     """
19     try:
20         rospy.wait_for_service(service_name, 2)
21     except (rospy.ServiceException, rospy.ROSException) as e:
22         rospy.logerr("Timeout and the Service was not available
       : " + str(e))
23         return RobotState()
```

```
24
25      try:
26          service_call = rospy.ServiceProxy(service_name, type)
27
28          if request_name == None:
29              response = service_call()
30          else:
31              request = request_name()
32              for key, value in req_args.items():
33                  #print("f{key} = {value}")
34                  method = setattr(request, key, value)
35              response = service_call(request)
36
37      except rospy.ServiceException as e:
38          rospy.logerr("Falied to call the Service")
39          return RobotState()
40
41      if should_return == 1:
42          if not response:
43              wait(3)
44          return response
```

Appendix B

Python Code with a white background

```
1
2  def Subscribe(topic_name, type, msg_args):
3      """Subscribe to a certain topic.
4
5      Parameters:
6      ......................
7      topic_name: str
8      type: srv
9      msg_args: list >> list of strings, which contains the
       arguments that we need to read from the topic.
10
11     Returns:
12     ......................
13     Return a list of the read values from each argument.
14     If we have only one argument, it returns the value of this
       argument only, not a list.
15     """
16
17     #rospy.init_node('FX_ROS_Subscriber')
18
19     try:
20         msg = rospy.wait_for_message(topic_name, type, 2)
21     except:
22         print('no topic found!')
23
24
25     value = []
26
27     if len(msg_args) == 1:
```

```
28          value = getattr(msg, msg_args[0])
29      else:
30          for i in msg_args:
31              value.append(getattr(msg, i))
32
33      return value
```